Дніпровський національний університет імені Олеся Гончара

Факультет прикладної математики

Кафедра математичного забезпечення ЕОМ

**ЗВІТ**

**з переддипломної практики**

**Тема:** *Розробка Android-додатку для слідкування за денною активністю та калоріями*

**Виконав:**

студент 3 курсу групи ПЗ-21у-1

спеціальності   
121 Інженерія програмного забезпечення

ОПП «Інженерія програмного забезпечення»

Микола ВОЙЦЕХОВ

(підпис) (Власне ім’я ПРІЗВИЩЕ)

**Керівник:**

Тетяна ЄМЕЛ’ЯНЕНКО

(підпис) (Власне ім’я ПРІЗВИЩЕ)

Національна шкала .

Кількість балів: .

Члени комісії . Тетяна ЄМЕЛ’ЯНЕНКО

(Підпис) (Власне ім’я ПРІЗВИЩЕ)

. Ольга МАЦУГА

(Підпис) (Власне ім’я ПРІЗВИЩЕ)

. Лілія БОЖУХА

(Підпис) (Власне ім’я ПРІЗВИЩЕ)

м. Дніпро

2024 р.

**ЗМІСТ**

[ВСТУП 3](#_Toc160032525)

[1 ОГЛЯДОВИЙ РОЗДІЛ 6](#_Toc160032526)

[1.1 Огляд існуючих фітнес додатків 6](#_Toc160032527)

[1.2 Огляд існуючих технологій та засобів мобільної розробки 7](#_Toc160032528)

[1.3 Опис обраних технологій та засобів розробки 8](#_Toc160032529)

[2 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 11](#_Toc160032530)

[2.1 Аналіз вимог до програмного забезпечення 11](#_Toc160032531)

[2.2 Опис архітектури програмного забезпечення 13](#_Toc160032532)

[2.3 Опис процесу проєктування програмного забезпечення 16](#_Toc160032533)

[3 ПРАКТИЧНІ РЕЗУЛЬТАТИ 20](#_Toc160032534)

[3.1 Приклади роботи програмного забезпечення 20](#_Toc160032535)

[ВИСНОВКИ 25](#_Toc160032536)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 26](#_Toc160032537)

# ВСТУП

У сучасному світі, де глобалізація і цифрові технології змінюють звичні підходи до подорожей, потреба в інноваційних рішеннях для планування та реалізації поїздок стає все більш вираженою. Одним з таких нововведень є коучсерфінг – соціальна практика, що передбачає безкоштовне короткострокове проживання у домівках місцевих жителів. Ця концепція не лише сприяє економії коштів під час подорожей, але й дозволяє глибше зануритися у культурне середовище регіону, забезпечуючи більш автентичний досвід.

Розробка мобільного додатку для коучсерфінгу відкриває перед подорожуючими нові можливості для взаємодії та обміну культурними цінностями. Такий додаток не тільки спростить процес пошуку господарів та гостей, але й наддасть користувачам зручні інструменти для комунікації, планування поїздок та обміну відгуками.

Актуальність дипломного проекту обумовлена зростаючою популярністю нестандартних форм туризму та збільшенням числа людей, які віддають перевагу подорожам з глибшим зануренням у культурне середовище, а не традиційним туристичним турам. Розробка мобільного додатку для коучсерфінгу відповідає сучасним тенденціям ринку туристичних послуг, де велику роль відіграють цифрові технології.

Метою дипломного проекту є розробка мобільного додатку, який надаватиме користувачам зручний інтерфейс, широкі функціональні можливості для пошуку житла та спілкування, а також високий рівень безпеки під час взаємодії між гостями та господарями.

Додаток повинен містити наступний функціонал:

1. Система профілю(реєстрація та авторизація);
2. Система рейтингу – після дати закінчення оренди, у обох сторін з’являється можливість виставити оцінку;
3. Система скарг­ ­– якщо на користувача за останній час прийшло багато скарг(більше 15 скарг за останній місяць) та його рейтинг менше ніж 2, то профіль буде заблоковано. Тільки адміністратор зможе розблокувати;
4. Система оренди – з можливістю наглядно на мапі подивитися, де знаходиться житло;
5. Пошук та сортування профіля та об’яв;
6. Система адміністратора;
7. Система сповіщень ­– орендодавець та орендар, за день та в день оренди будуть отримувати сповіщення.

Об'єктом дослідження є процеси взаємодії між користувачами мобільного додатку для коучсерфінгу, а предметом - функціональні та технічні аспекти мобільного додатку, що сприяють ефективній взаємодії між користувачами.

У ході дипломного проекту будуть розглянуті наступні аспекти:

1. Аналіз потреб користувачів мобільного додатку для коучсерфінгу.
2. Проектування архітектури та інтерфейсу мобільного додатку.
3. Розробка функціоналу для ефективного пошуку житла та спілкування між користувачами.
4. Забезпечення безпеки даних та взаємодій у мобільному додатку.
5. Тестування мобільного додатку та оцінка його ефективності на основі відгуків користувачів.

Дипломний проект передбачає комплексний підхід до розробки мобільного додатку для коучсерфінгу, включаючи аналіз потреб користувачів, проектування, програмування, тестування та впровадження. Результати проекту сприятимуть популяризації коучсерфінгу як альтернативного способу подорожей, а також розширенню можливостей для культурного обміну та спілкування між людьми з різних куточків світу.

Для досягнення мети було використано наступні інструменти:

1. Dart ­– мова програмування, яка використана для створення логіки додатку та взаємодії з компонентами;
2. PHP ­– був використаний для взаємодії мобільного додатку з сервером;
3. Flutter – фреймворк для створення графічного обліку додатку;
4. MySQL – реляційна система контролю баз даних, для зберігання інформації.

# 1 ОГЛЯДОВИЙ РОЗДІЛ

## **Огляд існуючих коучсерфінг** **додатків**

Couchsurfing є найбільш відомою та широко використовуваною платформою для коучсерфінгу, яка з'єднує мандрівників з місцевими жителями, які готові безкоштовно прийняти їх у своїх домівках. Сайт та мобільний додаток дозволяють користувачам шукати господарів, встановлювати з ними контакт та планувати свої подорожі.

Преваги**:**

* 1. Величезна глобальна спільнота, що легко дозволяє знайти господаря або гостя майже у будь-якому куточку світу.
  2. Різноманітні можливості для взаємодії та обміну культурним досвідом.
  3. Функції безпеки, включаючи верифікацію профілів, систему відгуків та рекомендацій.

Недоліки**:**

* 1. Необхідність платити за отримання статусу верифікованого користувача.
  2. Можливі складнощі з навігацією та використанням інтерфейсу сайту або додатку.
  3. Іноді зустрічаються скарги на проблеми з безпекою та конфіденційністю.

BeWelcome — це некомерційна платформа для коучсерфінгу, яка націлена на сприяння культурному обміну та гостинності по всьому світу. Вона зосереджена на створенні довірливих взаємовідносин між користувачами та пропонує зручні інструменти для пошуку місць проживання та спілкування.

Переваги**:**

* 1. Безкоштовне користування всіма функціями, включаючи верифікацію.
  2. Сильне почуття спільноти та підтримки серед користувачів.
  3. Відкритість та некомерційний характер платформи.

Недоліки**:**

* + 1. Менша база користувачів у порівнянні з Couchsurfing.
    2. Обмежені функціональні можливості та інструменти на сайті.

Trustroots створена з метою об'єднання людей, які поділяють цінності довіри, відкритості та взаємодопомоги. Це платформа спрямована на мандрівників, які шукають більш глибокого занурення у місцеві культури та спільноти.

Преваги**:**

* 1. Фокус на створенні міцних взаємин та довіри між користувачами.
  2. Простий та легкий у використанні інтерфейс.
  3. Повністю безкоштовний доступ до всіх функцій.

Недоліки**:**

* 1. Відносно мала кількість користувачів.
  2. Менше географічне покриття, особливо в менш популярних регіонах.
  3. Обмежені можливості для верифікації користувачів.

Кожна з цих платформ має свої унікальні особливості та відповідає певним потребам користувачів. Вибір платформи залежить від індивідуальних уподобань, цілей подорожі та бажаного рівня взаємодії з місцевими жителями та іншими мандрівниками.

## **1.2 Огляд існуючих технологій та засобів мобільної розробки**

**Android SDK (Software Development Kit)** є основним інструментарієм для розробки додатків під Android ОС. Він включає набір розробницьких інструментів, таких як дебагер, бібліотеки, емулятор пристрою, документацію, приклади коду, а також специфічні API для використання функцій Android пристроїв. Розробники можуть використовувати Android SDK через Android Studio, офіційне інтегроване середовище розробки (IDE) від Google, що спрощує процес розробки і тестування додатків.

Переваги:

* 1. Широкі можливості для розробки і тестування додатків.
  2. Підтримка від Google та великої спільноти розробників.
  3. Регулярні оновлення та вдосконалення інструментів.

Недоліки:

* 1. Вимагає часу на вивчення та адаптацію до постійно оновлюваних інструментів.
  2. Може виявитися складним для новачків у мобільній розробці.

**Java** — це універсальна, об'єктно-орієнтована мова програмування, яка використовується для розробки Android додатків. Є однією з найпопулярніших мов програмування для Android, завдяки своїй портативності, безпеці та легкості у використанні. Java дозволяє розробникам писати код, який може виконуватися на різних платформах без змін, що робить її ідеальною для мобільної розробки.

**Переваги:**

* 1. Велика спільнота та багато ресурсів для навчання.
  2. Висока переносимість коду між різними платформами.
  3. Сильна підтримка з боку розробників інструментів та бібліотек.

Недоліки:

* 1. В порівнянні з Kotlin, може мати більш високу складність і меншу виразність коду.
  2. Може вимагати більше ресурсів системи та часу на виконання.

Flutter — це відкритий фреймворк від Google для створення високоякісних нативних інтерфейсів на iOS та Android з єдиного кодової бази. Flutter використовує мову програмування Dart і забезпечує високу продуктивність додатків завдяки компіляції в нативний код.

Переваги:

* 1. Єдина кодова база для iOS та Android, що значно скорочує час розробки.
  2. Велика кількість готових до використання віджетів та висока продуктивність додатків.
  3. Гаряча перезавантаження (hot reload) для швидкого тестування змін у коді.

Недоліки:

* 1. Вимагає вивчення мови програмування Dart, яка менш відома порівняно з Java або Kotlin.
  2. Можуть виникати складнощі при інтеграції з деякими нативними бібліотеками або при використанні специфічних платформних можливостей.

Начало формы

**1.3 Опис обраних технологій та засобів розробки**

**Flutter** є відкритим фреймворком від Google для створення високоякісних нативних інтерфейсів на iOS і Android з єдиного коду. Він використовує мову програмування Dart і надає розробникам потужний набір інструментів та віджетів для створення інтуїтивно зрозумілих та візуально привабливих мобільних додатків.

Переваги:

* 1. **Швидка розробка**: Гаряче перезавантаження дозволяє миттєво побачити результати змін у коді.
  2. **Кросплатформність**: Єдиний кодовий базис для розробки додатків під Android та iOS.
  3. **Висока продуктивність**: Компіляція у нативний код забезпечує високу продуктивність додатків.
  4. **Гнучкий UI**: Великий вибір віджетів та можливість створення складних UI елементів.

Недоліки:

* 1. **Вивчення нової мови**: Необхідність вивчення Dart, хоча вона і легка для освоєння.
  2. **Розмір додатку**: Додатки на Flutter можуть бути більшими за розміром порівняно з нативними.
  3. **Обмежений доступ до деяких нативних функцій**: Може вимагати додаткової роботи для інтеграції специфічних нативних функцій.

PHP — це популярна загальноцільова скриптова мова програмування, яка активно використовується для розробки веб-сайтів. PHP легко інтегрується з HTML і використовується на серверній стороні, що робить його ідеальним для розробки динамічних веб-сайтів і веб-додатків.

Переваги:

* 1. Широка підтримка: Підтримується більшістю хостингових провайдерів.
  2. Гнучкість: Легко інтегрується з різними базами даних, такими як MySQL, PostgreSQL та іншими.
  3. Велика спільнота: Багато ресурсів для навчання, фреймворків та готових рішень.
  4. Простота вивчення: Легко почати для новачків у програмуванні.

Недоліки:

* 1. Безпека: Вимагає ретельного написання коду для уникнення вразливостей.
  2. Продуктивність: У деяких сценаріях може працювати повільніше порівняно з іншими мовами програмування.

**MySQL** — це найпопулярніша система управління базами даних SQL (СУБД), яка використовується у всьому світі для зберігання, обробки та отримання даних. Є відкритим програмним забезпеченням і часто використовується разом з PHP для розробки веб-додатків.

**Переваги:**

* 1. **Висока надійність та стабільність**: Масштабується для обробки великих обсягів даних.
  2. **Широка підтримка**: Інтегрується з багатьма мовами програмування та платформами.
  3. **Легкість використання**: Простий у вивченні синтаксис SQL для управління даними.
  4. **Безпека**: Включає розширені функції безпеки для захисту даних.

Недоліки:

* 1. **Обмеження в розподіленій обробці даних**: Не найкращий вибір для розподілених систем баз даних порівняно з NoSQL рішеннями.
  2. **Складність оптимізації**: Для великих та складних баз даних може вимагати додаткових зусиль для оптимізації продуктивності.

# 2 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## **2.1 Аналіз вимог до програмного забезпечення**

Під час аналізу вимог до програмного забезпечення коучсерфінга було проведено глибоке обговорення з потенційними користувачами та зацікавленими сторонами з метою зрозуміти їх потреби та очікування від додатку. Основні вимоги до програмного забезпечення визначено наступним чином:

1. Створення аккаунту: Користувач має можливість створити обліковий запис, редагувати та з необхідності видалити його.
2. Створення об’яви: Користувач має можливість додати нову об’яву, вказати адресу, кількість людей, дату заселення та виселення. З можливістю подальшого редагування та видалення.
3. Перегляд розміщених об’яв: При перегляді об’яв користувач має змогу сортувати за рейтингом, за датою створенням об’яви, за кількістю людей. Також шукати за країною або містом. Для більшої інформації о розташуванні житла, користувач має можливість переглянути на мапі та подивитися погодні умови на час прибування в обраному місці.
4. Система жалоб: Користувачі мають можливість шукати облікові записи інших користувачів та залишати на них скарги. Якщо на користувача прийшло більше 15 скарг за останній місяць та його рейтинг менше за 2, то його обліковий запис буде заблоковано. Для подальшого використання потрібно зв’язатися з адміністратором.
5. Система рейтингу: Після завершення дати об’яви для обох сторін стає доступна можливість оцінювання по 5 бальній шкалі.
6. Система сповіщення: За день та в день оренди користувачам буде приходити нагадування про оренду.
7. Система адміністратора: адміністратор має можливість видаляти об’яви та користувачів, переглядати жалоби, розблоковувати користувачів та відправляти їм повідомлення.

## **2.2 Опис архітектури програмного забезпечення**

Розробка мобільного додатку для коучсерфінгу, використовуючи Flutter для клієнтської частини, PHP як серверний бекенд, та MySQL для бази даних, вимагає чіткої архітектури для ефективної взаємодії між цими компонентами. Використовуючи архітектурний шаблон MVVM (Model-View-ViewModel), можна досягти високого рівня абстракції та незалежності компонентів, що спрощує розробку, тестування та підтримку додатку.

#### Клієнтська частина (Flutter):

1. **Модель (Model)**: Визначає структуру даних користувачів, господарів, житлових пропозицій, відгуків тощо. Модель включатиме класи для представлення цих даних і методи для взаємодії з сервером через RESTful API.
2. **Представлення (View)**: Включає інтерфейс користувача з екранами, діалогами, кнопками та іншими елементами управління. Використовуючи Flutter, можна створити гнучкий та відповідний UI, який буде однаково добре виглядати на різних пристроях.
3. **Модель представлення (ViewModel)**: Містить логіку представлення та бізнес-логіку, що відокремлюється від UI. ViewModel обробляє запити до моделі для отримання або зміни даних і підготовує ці дані для відображення в View.

#### Серверна частина (PHP):

1. **RESTful API**: Реалізує інтерфейс між клієнтською частиною та сервером, використовуючи HTTP-методи (GET, POST, PUT, DELETE) для взаємодії з базою даних. PHP скрипти будуть обробляти запити від клієнта, виконувати необхідні дії з базою даних (MySQL) та повертати дані у форматі JSON або XML.

#### База даних (MySQL):

1. **Схема бази даних**: Включає таблиці для користувачів, пропозицій житла, відгуків, бронювань тощо. Використовуючи відносинні зв'язки, можна ефективно організувати зберігання та доступ до даних.

### Переваги використання MVVM та RESTful API

1. **Розділення обов'язків**: MVVM спрощує управління кодом, розділяючи логіку інтерфейсу від бізнес-логіки, що полегшує розробку та підтримку.
2. Спрощення тестування: ViewModel може бути легко протестована незалежно від UI, що підвищує якість та надійність додатку.
3. **Масштабованість**: Архітектура дозволяє легко розширювати та додавати нові функції без великих змін у вже існуючому коді.
4. **Асинхронні операції**: З допомогою ViewModel можна ефективно управляти асинхронними даними, наприклад, завантаженням даних з сервера, забезпечуючи плавну роботу додатку.
5. **Стандартизація взаємодії**: Використання RESTful API стандартизує спосіб взаємодії між клієнтською та серверною частинами, роблячи комунікацію більш передбачуваною та легкою для розуміння.

Ця архітектура дозволяє розробникам мобільного додатку для коучсерфінгу створити надійний, масштабований та легко підтримуваний продукт, який може ефективно обробляти взаємодію користувачів з додатком, управління даними та взаємодію зі сторонніми сервісами або API.

## **2.3 Опис процесу проєктування програмного забезпечення**

Першим етапом було проведення ретельного аналізу вимог до програмного забезпечення. Цей процес включав у себе обговорення потреб та очікувань від додатку MealMinder з зацікавленими особами. На основі цього аналізу було сформульовано перелік основних функціональних та нефункціональних вимог до додатку.

Наступним кроком було створення архітектури системи для додатку MealMinder. Враховуючи вимоги до продукту та архітектуру MVVM, було розроблено моделі для даних таких як ActivityItem(Дані про активність), FoodItem(Дані про продукт в раціоні), FoodInfo(Інформація для пошуку продуктів), NewUser(Для передачі даних при реєстрації), PostFoodData(Для відправки даних з’їденого продукту на сервер) та деякі допоміжні моделі.

Також було створено моделі представлення для сторінок входу, реєстрації, вітальної сторінки, головної сторінки, пошуку продуктів та додавання продуктів.

Після створення архітектури додатку було побудовано модель бази даних, яка на даний момент складеться з 6 таблиць(Діаграму бази даних наведено на рисунку 2.1):

1. ActivityDayInfo: Модель представляє інформацію про активність користувача протягом дня. Містить зовнішні ключі до інформації про активності (ActivityInfo) та інформації про користувача (UserDayInfo).
2. ActivityInfo: Модель представляє типи активностей, які користувач може вводити. Має зв'язок один до багатьох з таблицею ActivityDayInfo.
3. DayFoodInfo: Ця модель відображає інформацію про їжу, яку спожив користувач протягом дня. Має зовнішні ключі до інформації про їжу (FoodInfo) та інформації про користувача (UserDayInfo).
4. FoodInfo: Модель містить дані про різні продукти та страви, їх назву та зображення. Має зв'язок один до багатьох з таблицею DayFoodInfo.
5. UserDayInfo: Ця модель відображає інформацію про день користувача, включаючи дату та інші характеристики. Має зв'язок один до багатьох з таблицею ActivityDayInfo та DayFoodInfo.
6. UserInfo: Модель представляє дані користувача, такі як ім'я, електронна пошта та пароль. Має зв'язок один до багатьох з таблицею UserDayInfo.

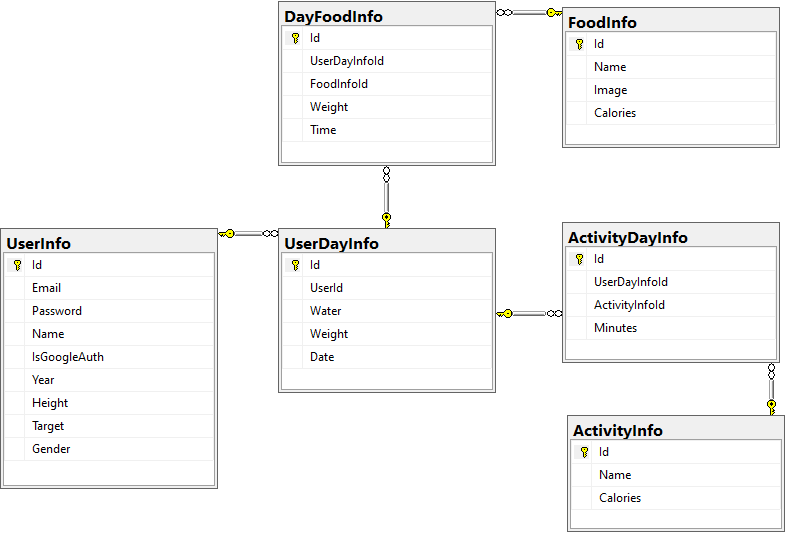


Рисунок 2.1 – Діаграма бази даних

Після побудови бази даних за допомогою ASP.NET Core Web API було реалізовано серверну частину додатку яка відповідає за взаємодію бази даних і мобільного додатку. Для підключення бази даних було використано Entity Framework, який дозволяє працювати з базою даних через сутності без використання прямих SQL команд, що пришвидшує та полегшує розробку веб сервісів. Для взаємодії з додатком було створено відповідні RESTful API ендпоінти, до яких звертаються моделі представлення у додатку.

Наступним кроком стало створення інтерфейсу користувача. Для цього використовувався інструмент Figma, який надає можливість створювати прототипи та дизайни інтерфейсу перед тим, як реалізувати їх у програмному коді. За допомогою цього інструмента було створено макети сторінок, які представлено на рисунку 2.2.

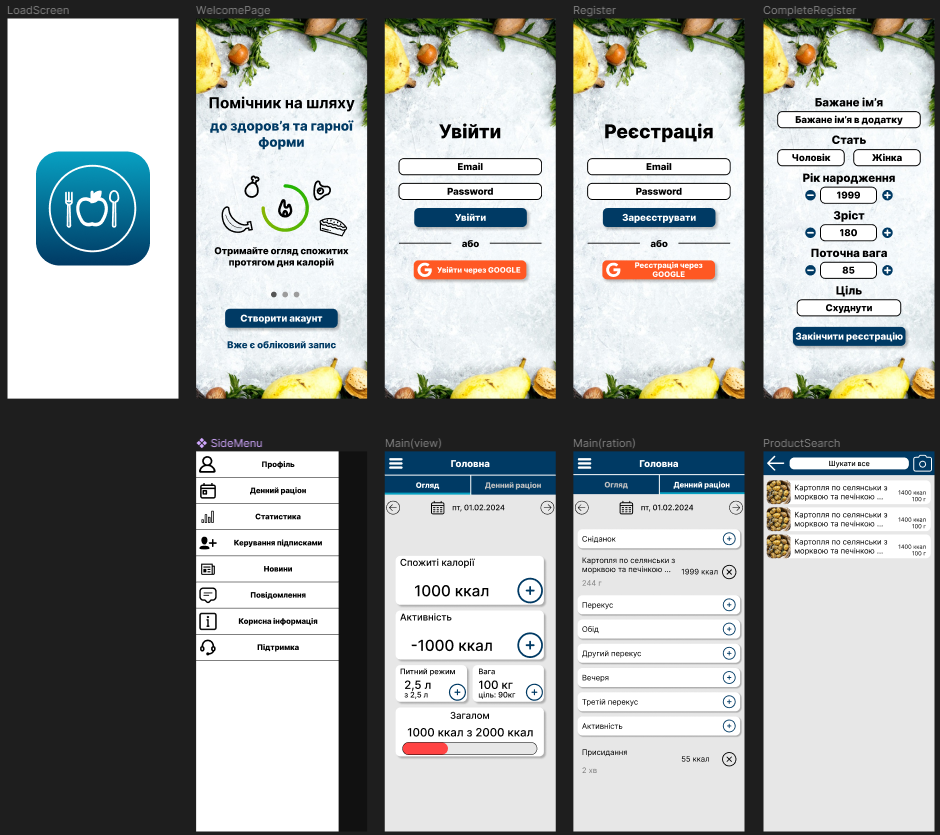


Рисунок 2.2 – Макети сторінок

Після створення макетів їх було реалізовано в додатку і підключено до відповідних моделей представлення, робочу сторінку представлено на рисунку 2.3.

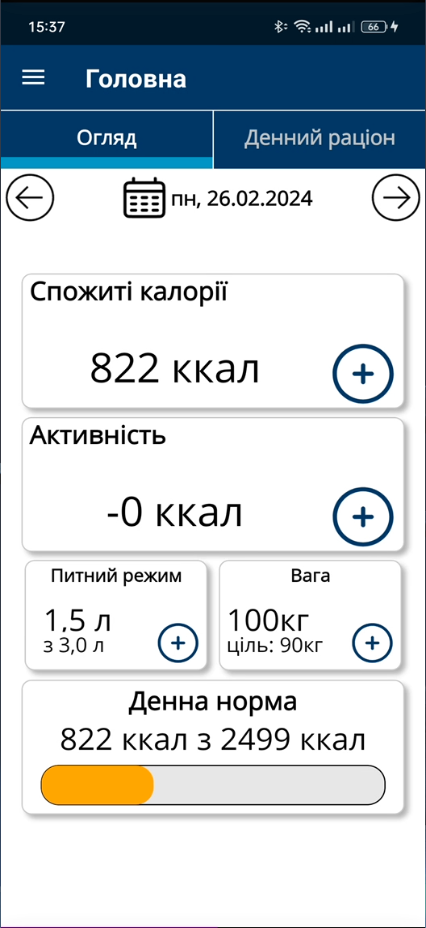


Рисунок 2.3 – Робоча сторінка додатку

# 3 ПРАКТИЧНІ РЕЗУЛЬТАТИ

## **3.1 Приклади роботи програмного забезпечення**

Сторінка привітання представлена на рисунку 3.1 з неї можна перейти до реєстрації або входу в акаунт.



Рисунок 3.1 – Сторінка привітання

При створенні аккаунту є два варіанти, перший через пошту і пароль, а другий використовує сервіс Google. Сторінка реєстрації представлена на рисунку 3.2.

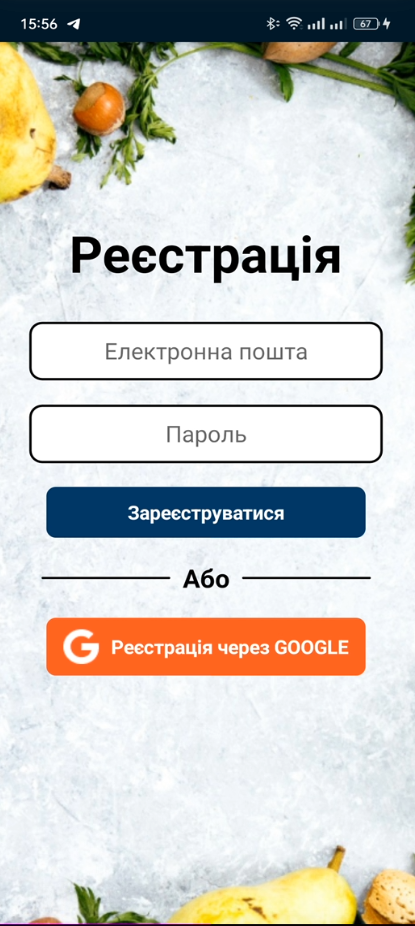


Рисунок 3.2 – Сторінка реєстрації

Наступна після реєстрації сторінка потрібна для заповнення даних про людину і завершення реєстрації, вона представлена на рисунку 3.3.

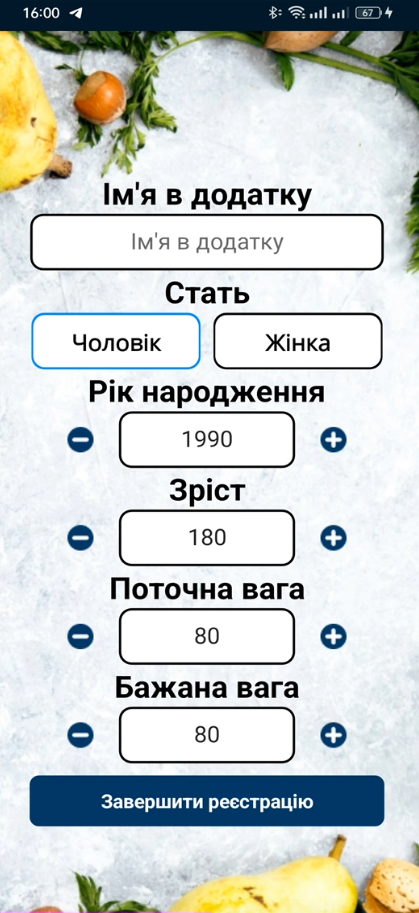


Рисунок 3.3 – Сторінка завершення реєстрації

Також сторінка входу представлена на рисунку 3.4, вона також має два варіанта входу в акаунт.

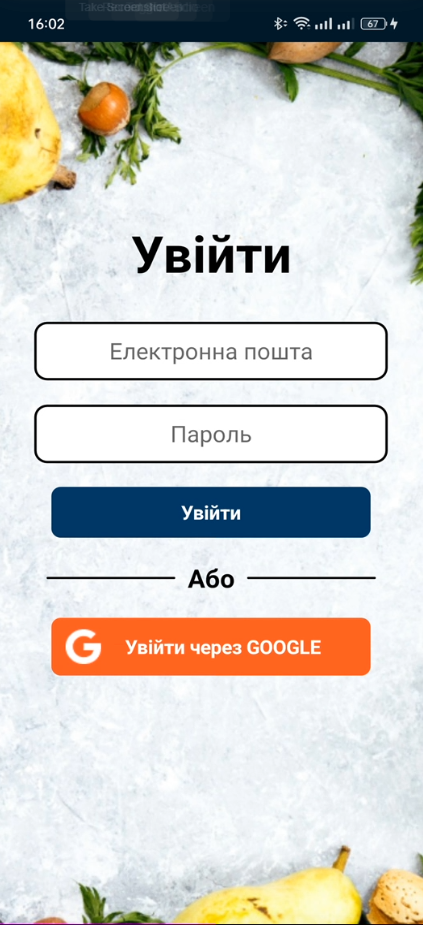


Рисунок 3.4 – Сторінка реєстрації

Після входу або реєстрації користувач переходить на головну сторінку додатка, на якій представлено стислі дані по денному раціону, активності, вазі та питному режиму, також є відповідні кнопки для швидких дій наприклад додати воду до питного режиму, або записати нову вагу. При додаванні активності або продуктів користувача буде переведено на відповідну сторінку пошуку з автоматичним збереженням часу додавання. У верхній частині сторінки є кнопки для детального перегляду раціону або переходу на інші дні. Вигляд головної сторінки у різні дні представлені на рисунках 3.5-3.6. На сторінці раціону можна детально переглянути які активності чи продукти ви вживали протягом дня або додати продукти у відповідні поля як план на день.

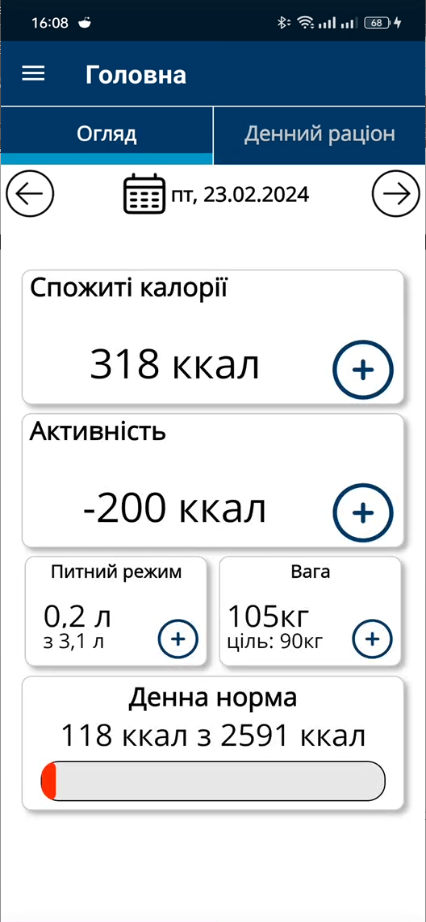
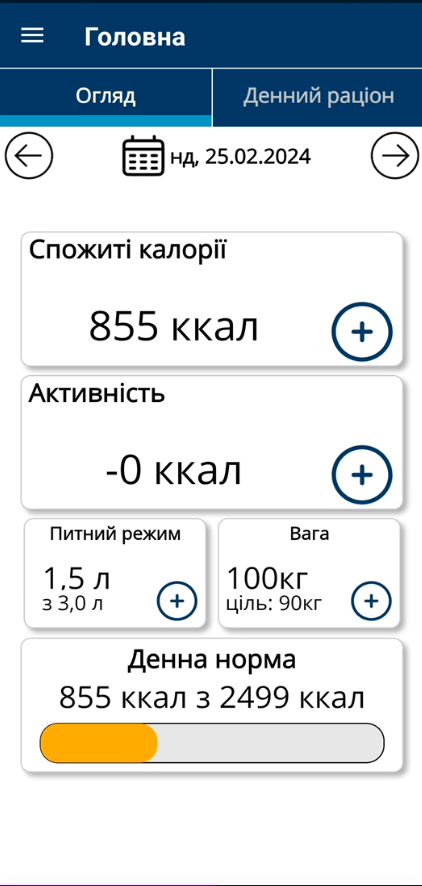


Рисунок 3.5 – Сторінка головного меню у різні дні

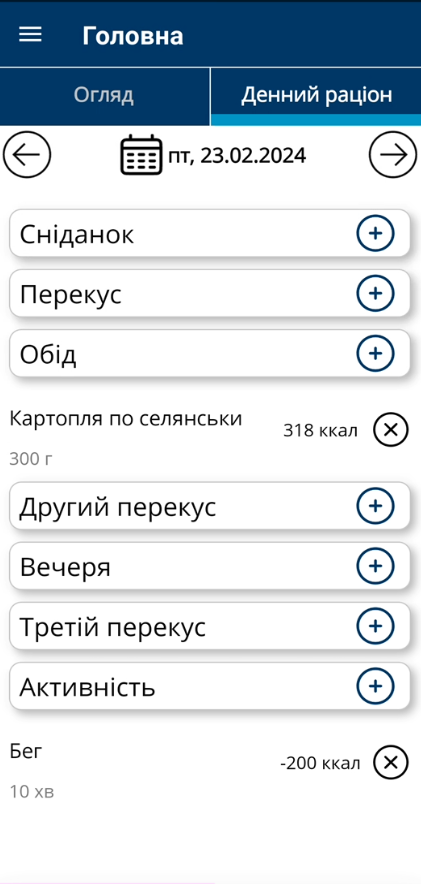
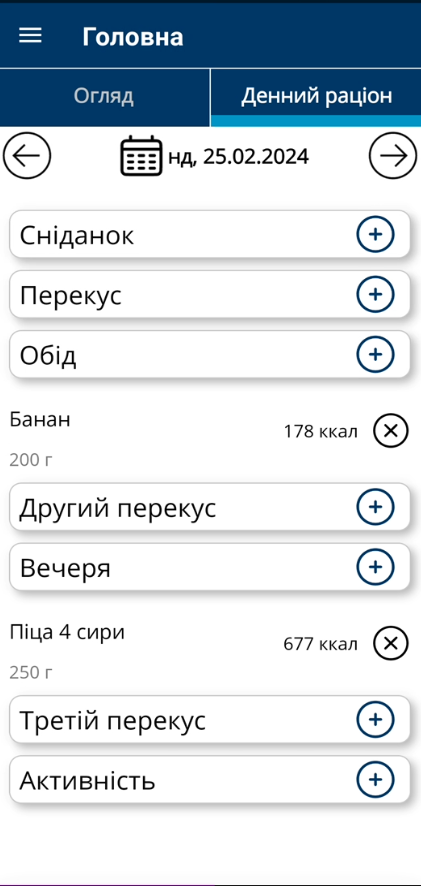


Рисунок 3.6 – Сторінка раціону у різні дні

Сторінка пошуку продуктів представлена на рисунку 3.7. На ній доступні функції пошуку по назві та відкриття камери для подальшого розпізнавання продукту по фотографії. При натисканні на продукт користувача буде переведено на сторінку додавання продукту, яка представлена на рисунку 3.8.

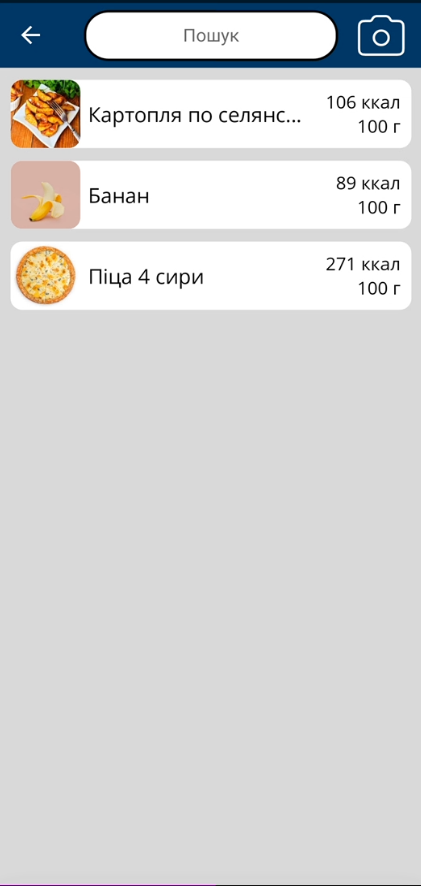


Рисунок 3.7 – Сторінка пошуку продуктів

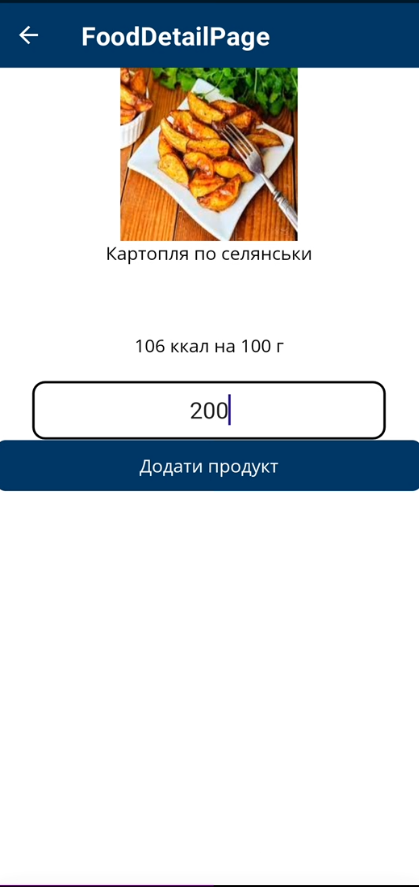


Рисунок 3.7 – Сторінка додавання продукту

# ВИСНОВКИ

Під час роботи над проєктом я отримав безцінний досвід у різних аспектах програмування та розробки програмного забезпечення. Розробка цього проєкту дозволила мені ознайомитися з новими для мене технологіями, такими як .NET Maui для крос-платформенної мобільної розробки, .NET Core Web API для створення веб-сервісів та TensorFlow для машинного навчання. А розробка архітектури додатку допомогла мені краще зрозуміти принципи роботи різних компонентів системи та їх взаємодію.

Також реалізація проєкту допомогла мені вдосконалити свої навички використання популярних дизайн-паттернів, таких як MVVM, а створення моделі штучного інтелекту дозволило мені вивчити та реалізувати алгоритми машинного навчання для розпізнавання їжі за фотографією, застосовуючи технології Transfer Learning, що дозволило мені отримати важливі навички в цій області.

Результатом роботи став додаток "MealMinder". Цей додаток дозволяє користувачам легко вести облік щоденного харчування, вправ та інших факторів, що впливають на їхнє здоров’я. Він надає широкий спектр функцій, включаючи підрахунок калорій, слідкування за питним режимом, вагою та іншими параметрами здоров'я.

Однією з головних переваг додатку є можливість розпізнавання їжі за фотографією за допомогою технології машинного навчання, що спрощує процес введення даних користувачами. Крім того, MealMinder забезпечує користувачів інтерактивними звітами та графіками для відстеження прогресу та стимулює спільноту до обміну рецептами та порадами.

Цей додаток є не лише зручним інструментом для ведення здорового способу життя, але й мотиваційною платформою для досягнення особистих фітнес-цілей. Він сприяє активному способу життя та сприяє поліпшенню загального стану здоров'я користувачів.

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

* 1. Android SDK. URL: <https://en.wikipedia.org/wiki/Android_SDK#:~:text=The%20Android%20SDK%20is%20a,comprehensive%20set%20of%20development%20tools>. (дата звернення: 28.02.2024).
  2. Java vs Kotlin. URL: <https://shoreteams.com/kotlin-vs-java/?gad_source=1&gclid=CjwKCAiA0PuuBhBsEiwAS7fsNXv7xyckYB23d8w4EaPHA7hfWlUmufcwexLI-S4KeqaPK_B4LrJCgBoCR2oQAvD_BwE> (дата звернення: 28.02.2024).
  3. Google Fit API. URL: <https://developers.google.com/fit> (дата звернення: 28.02.2024).
  4. C# Programming: What It Is, How It's Used + How to Learn It. URL: <https://www.coursera.org/articles/c-sharp> (дата звернення: 28.02.2024).
  5. What is .NET MAUI?. URL: <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-8.0> (дата звернення: 28.02.2024).